

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

SHOPIFY INC. AND SHOPIFY (USA),
INC.

Plaintiffs and Counterclaim
Defendants,

v.

EXPRESS MOBILE, INC.,

Defendant and Counterclaim
Plaintiff.

Civil Action No. 19-439-RGA

MEMORANDUM ORDER

This is a patent case about website design tools. Shopify, Inc. and its U.S. subsidiary seek declaratory judgment of non-infringement of the patents belonging to Express Mobile, Inc., which has filed a counterclaim of infringement. Currently before me is the issue of claim construction of various terms in those patents: U.S. Patent Nos. 6,546,397 ('397 patent), 7,594,168 ('168 patent), 9,063,755 ('755 patent), 9,471,287 ('287 patent), and 9,928,044 ('044 patent). The matter has been briefed (D.I. 117), and I heard oral argument on May 20, 2020 (D.I. 128). The terms are construed as set out below.

I. BACKGROUND

The specifications of the '397 and '168 patents are substantively identical, and the specifications of the '755, '287, and '044 patents are substantively identical.

The '397 patent family describes a browser-based tool for building webpages. ('397 patent at 1:6-8). The designer can make selections on various menus, and the system can preview

what the page will look like. (*Id.* at Abstract, 2:32-37). The data relating to the webpage is stored in an external database. (D.I. 120-1, Ex. 1, “Schmandt Decl,” ¶ 52). When an end user accesses the webpage, the user’s browser downloads a “run time engine” that is executed by a “virtual machine.” (*Id.* ¶ 53). The run time engine retrieves data from the external database and uses that data to display the webpage. (*Id.*). According to the specification, this technique is superior to “conventional web site construction tools,” which were not designed for “serious multimedia applications” and can be “remarkably slow and inefficient.” (’397 patent at 1:11-21).

The ’755 patent family is specifically directed to displaying content on mobile devices, such as smartphones. (’755 patent at Abstract). The inventions enable designers to integrate third-party web services into their content. (Schmandt Decl ¶ 55). The patent family discloses an authoring tool that allows designers to customize the display for receiving inputs and displaying outputs of web services. (*Id.*). Additionally, the patents disclose the concept of providing two sets of code to a user’s device: an “Application” and a “Player.” (*Id.* ¶ 58). The advantage of this system is that “all of the device-dependent programming is provided to the device only once (or possibly for some small number of upgrades), permitting a smaller Application, which is the same for each device.” (’755 patent at 5:52-55).

II. LEGAL STANDARD

“It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (internal quotation marks omitted). When construing patent claims, a court considers the literal language of the claim, the patent specification, and the prosecution history. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 977–80 (Fed. Cir. 1995) (en banc),

aff'd, 517 U.S. 370 (1996). Of these sources, “the specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.” *Phillips*, 415 F.3d at 1315 (internal quotation marks omitted).

“[T]he words of a claim are generally given their ordinary and customary meaning. . . . [Which is] the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.” *Id.* at 1312–13 (citations and internal quotation marks omitted). “[T]he ordinary meaning of a claim term is its meaning to [an] ordinary artisan after reading the entire patent.” *Id.* at 1321 (internal quotation marks omitted). “In some cases, the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words.” *Id.* at 1314.

When a court relies solely upon the intrinsic evidence—the patent claims, the specification, and the prosecution history—the court’s construction is a determination of law. *See Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 135 S. Ct. 831, 841 (2015). The court may also make factual findings based upon consideration of extrinsic evidence, which “consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises.” *Phillips*, 415 F.3d at 1317–19 (internal quotation marks omitted). Extrinsic evidence may assist the court in understanding the underlying technology, the meaning of terms to one skilled in the art, and how the invention works. *Id.* Extrinsic evidence, however, is less reliable and less useful in claim construction than the patent and its prosecution history. *Id.*

III. CONSTRUCTION OF DISPUTED TERMS

1. “virtual machine” (’397 and ’168 patents)
 - a. *Express Mobile’s Construction*: “abstract machine emulated in software”
 - b. *Shopify’s Construction*: “abstract machine that is emulated in software and that executes intermediate code in the instruction set of that machine”
 - c. *Court’s Construction*: “software that emulates a physical machine”

The parties agree that a “virtual machine” is a software program that emulates the functions of a physical machine. Shopify seeks the additional limitation that the machine “executes intermediate code in the instruction set of that machine.”

There is little intrinsic evidence on the meaning of this term. The specification refers to a “virtual machine” only once, identifying a “Java Virtual Machine” in the descriptions of drawings. (’397 patent at 35:34-38). According to Shopify, a Java virtual machine (JVM) uses Java bytecode, which is a type of intermediate code. (D.I. 117 at 10). That may be so. But the “Java Virtual Machine” mentioned in the specification is clearly just one possible embodiment, not the invention itself. It is improper to “import a feature from a preferred embodiment into the claims.” *Acumed LLC v. Stryker Corp.*, 483 F.3d 800, 805 (Fed. Cir. 2007).

In *X.Commerce, Inc. v. Express Mobile, Inc.*, the court rejected the similar argument that this claim term is limited to “compiled code.” WL 10704439, at *3 (N.D. Cal. Sept. 12, 2018). The court concluded that although “the specification was drafted to explain the invention in the context of the JVM, the then-dominant virtual machine in the particular technological context, . . . [i]t simply does not follow . . . that the patent claims are limited to virtual machines that execute

compiled code.” I agree, and I see no reason that “executes intermediate code in the instruction set of that machine” should be any different. I therefore reject Shopify’s proposed construction.

I construe this term as “software that emulates a physical machine” because I believe it will be easier for a jury to readily understand than “abstract machine emulated in software,” which could lead to unnecessary confusion around the word “abstract.” Express Mobile’s counsel indicated at oral argument that Express Mobile had no objection to this construction. (D.I. 128 at 12:18-23).

2. “run time engine” (’397 and ’168 patents)

- a. *Express Mobile’s Construction*: “file that is executed at runtime that facilitates the retrieval of information from the database and generates commands to display a web page or website”
- b. *Shopify’s Construction*: “file that is executed at runtime that reads information from the database and generates virtual machine commands to display a web page or website”
- c. *Court’s Construction*: “file that is executed at runtime that reads information from the database and generates commands to display a web page or website”

The parties have two disputes about this term. First, they dispute whether the run time file “reads” information or merely “facilitates the retrieval of information.” Second, they dispute whether the run time file generates “virtual machine commands” or just “commands.”

As to the first dispute, the specification repeatedly describes the run time engine as “read[ing]” information from the database. (*See* ’397 patent at 5:52-57, 45:44-57). Admittedly, these references appear in descriptions of embodiments, but there is no intrinsic support for Express Mobile’s argument that the run time engine “facilitates the retrieval of information.” That wording is also needlessly vague. *See X.Commerce*, 2018 WL 10704439, at *3 (finding that

this same proposed construction “introduces unwarranted ambiguity not grounded in anything expressly described in the specification.”).

As to the second dispute, there is less intrinsic support for Shopify’s proposal that the file must generate “virtual machine commands.” Shopify notes that during prosecution, the applicant mentioned “virtual machine commands” while distinguishing the invention from the prior art. That feature, however, was not presented as a distinction. Rather, the applicant was arguing the invention was novel because it disclosed a run time engine and a separate database of user settings. (D.I. 69-2, Ex. 10 at 5 (“In contrast, the claimed invention generates web pages using two features: a run time engine and a database of user settings.”)). Thus, I find that the file executes “commands,” not necessarily “virtual machine commands.”

3. “Application” (’755, ’287, and ’044 patents)

- a. *Express Mobile’s Construction*: “device-independent software code containing instructions for a device”
- b. *Shopify’s Construction*: “device-independent code that is generated by the authoring tool, is partitioned from the Player, and is interpreted or executed by the Player”
- c. *Court’s Construction*: “device-independent code which contains instructions for a device and which is separate from the Player”

The parties agree (and the specification is explicit) that the “Application” is “device-independent” software code. Shopify argues the Application must also be: 1) “generated by the authoring tool;” 2) “partitioned from the Player;” and 3) “interpreted or executed by the Player.”

For the first issue, Shopify points to a statement by the applicant during prosecution that the authoring tool generates the Application and the Player. (D.I. 117 at 68). That statement, however, is explaining a drawing from the application and is not clearly limiting the invention itself. (D.I. 69-2, Ex.19 at 8-9). The specification itself is vague on how the Application is

generated. Claim 26 of the '755 patent depends on claim 23 and adds only the limitation that the Application and Player are “generated using an authoring tool.” This narrower claim suggests a broader reading of “Application” in claim 23 because otherwise the claims would be identical. While claim differentiation “is not a hard and fast rule,” *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1374 (Fed. Cir. 2014), there is no contrary evidence in the specification here that would override that presumption. Therefore, I decline to require that the Application be “generated by the authoring tool.”

Second, Shopify argues that the Application is “partitioned from the Player.” During prosecution, the applicant touted separation of the Application and Player as a novel innovation over the prior art. (*See* D.I. 69-2, Ex. 18 at 12, 13) (“[T]he claimed invention, in contrast, operates by partitioning the code required for functionality into device-independent code and device-dependent code.”) (D.I. 69-2, Ex. 19 at 8) (“[P]artitioning code for accessing web services into an Application and a Player has an advantage for maintaining websites.”) (D.I. 69-2, Ex. 15 at 11) (acknowledging that a prior art reference, “McCain,” discloses an authoring tool that provides code stored in a database, but arguing that “[t]here is no teaching or suggestion in McCain of an authoring tool that provides two separate codes.”). Glenn E. Weadock, Express Mobile’s expert, acknowledged during his deposition that “creating a separate application and player—having two different things, having an application and a player that are distinct from each other has certain benefits.” (D.I. 120-1, Ex. 3 at 175:24-176:16).

Express Mobile does not dispute that the Application must be “separate” or “distinct” from the Player. (D.I. 117 at 72; D.I. 128 at 72:5-7). Rather, Express Mobile’s concern is that the word “partitioned” could artificially limit the term. Express Mobile points to an embodiment in

the specification in which the Player and Application are integrated on a single server (although they presumably remain separate sets of code). ('755 patent at 7:30-3). Shopify does not argue there must be some physical barrier between the Application and the Player. Although the applicant used the word “partitioned” during prosecution, I will not adopt that word here to avoid a construction that could be read as excluding an embodiment described in the specification. *See MBO Labs., Inc. v. Becton, Dickinson & Co.*, 474 F.3d 1323, 1333 (Fed. Cir. 2007) (“[A] claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct.”). Given the applicant’s clear statements during prosecution though, I decide the Application must be “separate” from the Player.

Finally, Shopify argues the Application is “interpreted or executed by the Player.” In support of this construction, Shopify cites the specification’s disclosure that the “Player interprets or executes the Application” ('755 patent at 6:6-7) and the applicant’s statement during prosecution that “the Player interprets the Application” (D.I. 69-2, Ex. 15 at 9). Both these statements, however, appear to be describing embodiments, not the invention itself. The specification statement is in a section labeled, “Mode(s) for Carrying Out the Invention,” which describes various drawings. For the prosecution statement, the previous paragraph begins, “The Present Application includes embodiments . . . ,” which suggests the subsequent statement about the Player interpreting the Application is a description of one of those embodiments featured in the application.

Express Mobile proposes that the Application “contain[s] instructions for a device.” Given that Shopify’s briefing does not address this point, and it seems to make sense, I will include it in the construction.

4. “Player” (’755, ’287, and ’044 patents)
 - a. *Express Mobile’s Construction*: “Software code that facilitates the execution of an application on a device”
 - b. *Shopify’s Construction*: “an executable file of device-specific instructions for the processor of the device that is generated by the authoring tool, is partitioned from the Application, and interprets or executes the Application”
 - c. *Court’s Construction*: “device-specific code which contains instructions for a device and which is separate from the Application”

Based on the same intrinsic evidence discussed above for the “Application,” I conclude the “Player” is not necessarily “generated by the authoring tool,” nor must it “interpret[] or execute[] the Application.” And for the same reasons as above, the Player is “separate” from the Application.

Shopify argues the Player is a file of “device-specific” instructions. The parties agree (as discussed above) that the “Application” is “device-independent” code, but they disagree on whether the corresponding “Player” is “device-specific” code. The Abstract discloses, “Devices are provided with Players specific to each device and Applications that are device independent.” (’755 patent at Abstract). The Federal Circuit has explained that “[w]hile a statement in the Abstract may operate as a clear expression of manifest exclusion,” courts should be cautious not to give the Abstract too much weight in claim construction because “[t]his section of a patent speaks generally to the invention and, much like the syllabus of an opinion, sets forth general information about the document’s content.” *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1121 (Fed. Cir. 2004).

The specification discloses numerous embodiments that feature a Player with device-specific instructions. *See* ’755 patent at 5:8-24; 23:43-46; 33:12-15; 33:26-28. Shopify also

points to the applicant's statements during prosecution referring to a device-specific player.

These statements are worth quoting at length:

The present invention relates to the field of displaying information on a device, such as a mobile device, over a network. One embodiment is an authoring tool that generates files which together provide content over a network. The files include a Player (sometimes referred to herein as a "first code") specific to each device (that is, the code is "device-dependent") and an Application (sometimes referred to herein as a "second code") that is device-independent. The device-dependent Player is activated by a web browser or other software or through a signal to device by a special telephone numbers (sic), such as a short code (see, for example, Paragraph [0049]).

In one aspect of the present invention, the device-independent Application and device-dependent Player are both provided to a device, such as a smartphone. Both codes are executed or interpreted on the same device, which communicates with a web service.

(D.I. 69-2, Ex. 19 at 8).

The present patent application consistently use the words "Application" (with a capital A) and "Player" (with a capital P) to refer to code that is provided to devices for accessing web services, where an Application is **device-dependent code** and a Player is **device-dependent code**.

(*Id.* at 9-10, n.1) (emphasis in original). Notwithstanding the obvious typo in this sentence, its intended meaning is clear.

Since this code [described in prior art, McCain] operates on devices to access the web, and includes user interfaces for such devices, the code is by necessity device-dependent, and is not equivalent to the claimed "Application." More specifically, since the code is generated to run on a specific device, such as server 34, McCain is teaching the generation of a Player.

(*Id.* at 12).

There is no teaching or suggestion in McCain of an authoring tool that provides two separate codes: a device-dependent code (such as the claimed Player) and device-independent code (such as the claimed Application).

(D.I. 69-2, Ex. 15 at 11)

There is no teaching or suggestion in McCain of using such a solution database in an authoring tool to generate both a device-dependent and device-dependent codes, as claimed.

(*Id.* at 12). Notwithstanding the obvious typo in this sentence, its intended meaning is clear.

Express Mobile responds that the specification explicitly discloses an embodiment where the Player is not device-specific:

In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that manages resources efficiently, is extensible, supports a robust application model, and has no device specific dependencies.

(’755 patent at 1:55-67).

In its briefing, Shopify argues this reference to “no” device specific dependencies must have been a typo. (D.I. 117 at 76). Given the frequency of typos on this point in the prosecution history (as in the two excerpts quoted above), this argument has a plausible basis in fact. At oral argument, Shopify’s counsel offered another explanation: “no device specific dependencies” is referring to “language,” not to the “Player.” (D.I. 128 at 113:9-17). Presumably, this argument is based on the quoted sentence being able to be read as either:

In one embodiment, one of the codes is a Player, which is a thin client architecture that (1) operates in a language that manages resources efficiently, (2) is extensible, (3) supports a robust application model, and (4) has no device specific dependencies.

Or,

In one embodiment, one of the codes is a Player, which is a thin client architecture that operates in a language that (1) manages resources efficiently, (2) is extensible, (3) supports a robust application model, and (4) has no device specific dependencies.

The argument has grammatical appeal. But whether a POSITA would understand it one way or the other for non-grammatical reasons has not been fleshed out in the briefing or at oral argument.

This debate over whether the Player is necessarily device-specific implicates a limited number of the asserted claims. Claims 1, 12, and 27 of the '755 patent and claim 1 of the '287 patent all explicitly define the "Player" as "device-dependent code" (which I understand to mean the same thing as "device-specific" code). Claims 1 and 15 of the '044 patent, however, do not define the "player" (with a lower-case "p" in those claims). The '044 patent shares the specification of the '755 and '287 patents, and it was prosecuted as a continuation of the '755 patent application.

I conclude that a person of ordinary skill in the art, looking at this intrinsic evidence as a whole, would understand the Player to be "device-specific." The single reference to "no device specific dependencies" is not enough to override the repeated references in the prosecution history and the specification to the "device-specific" or "device-dependent" Player. During prosecution, the applicant represented the "device-independent" Application and the "device-dependent" Player as central to the invention. In particular, the applicant cited these features as the key innovations over the prior art of McCain. (*See* D.I. 69-2, Ex. 15 at 11, 12). "[P]rosecution history may be critical in interpreting disputed claim terms because it contains the complete record of all the proceedings before the Patent and Trademark Office, including any express representations made by the applicant regarding the scope of the claims." *Sunovion Pharm., Inc. v. Teva Pharm. USA, Inc.*, 731 F.3d 1271, 1276 (Fed. Cir. 2013).

The Federal Circuit has held that even when “prosecution history statements do not rise to the level of unmistakable disavowal, they do inform the claim construction. For example, an applicant’s repeated and consistent remarks during prosecution can define a claim term by demonstrating how the inventor understood the invention.” *Personalized Media Commc’ns, LLC v. Apple Inc.*, 952 F.3d 1336, 1340 (Fed. Cir. 2020) (cleaned up). Here, the applicant made “repeated and consistent remarks” indicating that the Player was device-specific. A person of ordinary skill would rely on those remarks over the single arguably ambiguous specification reference to “no device specific dependencies,” which could be read as referring to “language.”

Regardless, the prosecution history evidence also meets the higher standard of prosecution history disclaimer. “[A] patentee may limit the meaning of a claim term by making a clear and unmistakable disavowal of scope during prosecution. This may occur, for example, when the patentee explicitly characterizes an aspect of his invention in a specific manner to overcome prior art.” *Purdue Pharma L.P. v. Endo Pharm. Inc.*, 438 F.3d 1123, 1136 (Fed. Cir. 2006). Here, the patentee distinguished the prior art McCain by pointing to the “device-dependent code (such as the claimed Player) and device-independent code (such as the claimed Application).” (D.I. 69-2, Ex. 15 at 11). This statement, along with the other consistent representations to the patent examiner, constitute “clear and unmistakable disavowal” of a device-independent Player.

Shopify further argues that the Player must be an “executable file” and that it must contain “instructions for the processor of the device.” The intrinsic evidence does not support these limitations. Shopify cites a reference in the specification that the Player code contains “instructions for the processor,” but this reference is to a single embodiment. (’755 patent at 5:9-

12). Similarly, Shopify cites isolated statements in the prosecution history about “executable” code. (D.I. 69-2, Ex. 18 at 9; Ex. 15 at 10). These references are not the kind of repeated and consistent remarks that would narrow the definition of the term. According to Express Mobile’s expert, Glenn E. Weadock, “if (as the patent describes) the player can ‘extend the operating system,’ it could consist of multiple files called by the operating system, none of which is executable on its own . . . At most, the specification suggests that the Player contains instructions that, eventually, are executable.” (D.I. 118-1, Ex. 2 ¶ 106). Thus, I reject these proposed limitations. As with Application though, I include that the code “contains instructions for a device” (although not necessarily the processor of the device) because that language does not appear to be in dispute. (*See* D.I. 117 at 77) (Express Mobile acknowledging that the Player contains instructions).

I understand “device-specific” and “device-dependent” to be interchangeable terms. I think it might help a jury to use the former rather than the latter. When listening to testimony, “device-specific” is easier to differentiate from “device-independent” than “device-dependent” is.

5. “device-dependent code” (’755 and ’287 patents)
 - a. *Express Mobile’s Construction*: “code that is aware of device type or device-specific functions”
 - b. *Shopify’s Construction*: “executable code that differs depending on the device platform”
 - c. *Court’s Construction*: “code that is specific to the operating system, programming language, or platform of a device”

6. “device-independent code” (’755 and ’287 patents)
 - a. *Express Mobile’s Construction*: No construction necessary
Alternative: “code that is not aware of or dependent on device type or device-specific functions”
 - b. *Shopify’s Construction*: “code that is the same for every device platform”
 - c. *Court’s Construction*: “code that is not specific to the operating system, programming language, or platform of a device”

The claims of the ’755 and ’287 patents explicitly define the Application as “device-independent code” and the Player as “device-dependent” code. Express Mobile suggests that device-dependent code is “aware” of device type or device-specific functions (and consequently, device-independent code is “not aware”). The specification does describe software in one embodiment as being “aware” of device functions. (’755 patent at 34:4-11). I do not think it would be helpful though to have the jury ponder the metaphysical question of whether software is “aware” of anything.

The specification at one point defines “device-specific” routines as “codes that are specific to the operating system, programming language, or platform of specific devices.” (*Id.* at 3:58-62). This definition is consistent with how the applicant discussed device-dependent code during prosecution: “If a new device comes on the market, or if it is found that the device-dependent code for a specific device needs an update, a new Player is developed and provided to those specific devices.” (D.I. 69-2, Ex. 19 at 8). This definition is preferable to Shopify’s proposed construction, which adds the unnecessary word “executable” and requires that the code differ based only on the device “platform.” Thus, I construe “device-dependent code” to mean: “code that is specific to the operating system, programming language, or platform of a device” (removing only a redundant “specific” from the specification language).

For “device-independent code,” I adopt the inverse of this definition: “code that is not specific to the operating system, programming language, or platform of a device.” Shopify proposes that the code must be “the same for every device platform.” This definition is too narrow because the code could theoretically be “device-independent” while not being the same for all devices. That is, the code could differ based on some other factors that are “independent” of the device platforms.

IV. CONCLUSION

The Joint Claim Construction Brief (D.I. 117) had twenty-six disputed terms. Before the Markman hearing, I directed that the parties reduce the number of terms that would be argued to no more than ten, and I asked to have them argued in order of importance. (D.I. 126). Despite lengthy argument, only four terms were addressed. (D.I. 128 at 122). I have resolved the four terms. I also resolved two others that were closely related to the ones that were argued. (*See* D.I. 128 at 124). The ones that have not been resolved will not be resolved unless the parties dispute them at the time of the summary judgment briefing. (*See* D.I. 126).

The parties should submit within five days a proposed order, suitable for submission to a jury, construing the terms as indicated.

Entered this 23rd day of June, 2020.

/s/ Richard G. Andrews
United States District Judge